

# **An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Make a Big Difference?**

---

Ryan Giordano (rgiordan@mit.edu)<sup>1</sup>  
Spring 2022

---

<sup>1</sup>With coauthors Rachael Meager (LSE) and Tamara Broderick (MIT)

# Dropping data: Mexico Microcredit

**Example:** Angelucci et al. [2015], a randomized controlled trial study of the efficacy of microcredit in Mexico based on  $N = 16,560$  data points. A regression was run to estimate the average effect of microcredit.

---

**Original result:** Treatment effect statistically insignificant at 95%.

**Policy implication:** Disinvest in microcredit initiatives.

---

# Dropping data: Mexico Microcredit

**Example:** Angelucci et al. [2015], a randomized controlled trial study of the efficacy of microcredit in Mexico based on  $N=16,560$  data points. A regression was run to estimate the average effect of microcredit.

---

**Original result:** Treatment effect statistically insignificant at 95%.

**Policy implication:** Disinvest in microcredit initiatives.

---

**Data dropping:** Can produce both positive and negative statistically significant results dropping no more than 15 data points ( $< 0.1\%$ ).

**Policy implication:** Run a higher-powered study (not just larger  $N$ ).

---

# Dropping data: Mexico Microcredit

**Example:** Angelucci et al. [2015], a randomized controlled trial study of the efficacy of microcredit in Mexico based on  $N = 16,560$  data points. A regression was run to estimate the average effect of microcredit.

---

**Original result:** Treatment effect statistically insignificant at 95%.

**Policy implication:** Disinvest in microcredit initiatives.

---

**Data dropping:** Can produce both positive and negative statistically significant results dropping no more than 15 data points ( $< 0.1\%$ ).

**Policy implication:** Run a higher-powered study (not just larger  $N$ ).

---

Cannot find influential subsets by brute force!

**We provide a fast, automatic tool to approximately identify the most influential set of points.**

- Why and when might you care about sensitivity to data dropping?
- How do we identify influential sets? When is our method accurate?  
(A formalization of the problem and the class of estimators we study.)
- Examine real-life examples of analyses: some sensitive, some not.  
(The results may defy your intuition.)
- What kinds of analyses are sensitive to data dropping?  
(Comparison to standard errors, gross errors, and how to mitigate.)

# Dropping data: Motivation

When do you care that you can **reverse your conclusion** by removing a **small proportion** of your data?

# Dropping data: Motivation

When do you care that you can **reverse your conclusion** by removing a **small proportion** of your data?

Not always! But sometimes you surely do, especially when you want to **generalize to unseen, systematically different populations**.

Suppose you have a farm, and want to know whether your average yield is  $> 170$  bushels per acre. At harvest, you measure 200 bushels per acre.

# Dropping data: Motivation

When do you care that you can **reverse your conclusion** by removing a **small proportion** of your data?

Not always! But sometimes you surely do, especially when you want to **generalize to unseen, systematically different populations**.

Suppose you have a farm, and want to know whether your average yield is  $> 170$  bushels per acre. At harvest, you measure 200 bushels per acre.

- Scenario one:  $> 170$  bushels per acre means you make a profit.
  - Don't care about sensitivity to small subsets.



# Dropping data: Motivation

When do you care that you can **reverse your conclusion** by removing a **small proportion** of your data?

Not always! But sometimes you surely do, especially when you want to **generalize to unseen, systematically different populations**.

Suppose you have a farm, and want to know whether your average yield is  $> 170$  bushels per acre. At harvest, you measure 200 bushels per acre.

- Scenario one:  $> 170$  bushels per acre means you make a profit.
  - Don't care about sensitivity to small subsets.
- Scenario two: Want to recommend methods to a distant friend.
  - Might care about sensitivity to small subsets!

# Dropping data: Motivation

When do you care that you can **reverse your conclusion** by removing a **small proportion** of your data?

Not always! But sometimes you surely do, especially when you want to **generalize to unseen, systematically different populations**.

Suppose you have a farm, and want to know whether your average yield is  $> 170$  bushels per acre. At harvest, you measure 200 bushels per acre.

- Scenario one:  $> 170$  bushels per acre means you make a profit.
  - Don't care about sensitivity to small subsets.
- Scenario two: Want to recommend methods to a distant friend.
  - Might care about sensitivity to small subsets!

Specifically, often in statistical applications:

- Policy population is different from analyzed population,
- Small fractions of data are missing not-at-random,
- We report a convenient summary (e.g. mean) of a complex effect.

# Formalizing the question.

## Example: Least squares

# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\hat{\theta} := \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - \theta^T x_n)^2$$

$$\Leftrightarrow \sum_{n=1}^N (y_n - \hat{\theta}^T x_n) x_n = 0.$$

# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\hat{\theta} := \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - \theta^T x_n)^2$$
$$\Leftrightarrow \sum_{n=1}^N (y_n - \hat{\theta}^T x_n) x_n = 0.$$

Make a qualitative decision using:

- A particular component:  $\hat{\theta}_k$
- The end of a confidence interval:  $\hat{\theta}_k + \frac{1.96}{\sqrt{N}} \hat{\sigma}(\hat{\theta})$

# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\hat{\theta} := \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - \theta^T x_n)^2$$

$$\Leftrightarrow \sum_{n=1}^N (y_n - \hat{\theta}^T x_n) x_n = 0.$$

Make a qualitative decision using:

- A particular component:  $\hat{\theta}_k$
- The end of a confidence interval:  $\hat{\theta}_k + \frac{1.96}{\sqrt{N}} \hat{\sigma}(\hat{\theta})$

## General setup: Z-estimators

# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\hat{\theta} := \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - \theta^T x_n)^2$$
$$\Leftrightarrow \sum_{n=1}^N (y_n - \hat{\theta}^T x_n) x_n = 0.$$

Make a qualitative decision using:

- A particular component:  $\hat{\theta}_k$
- The end of a confidence interval:  $\hat{\theta}_k + \frac{1.96}{\sqrt{N}} \hat{\sigma}(\hat{\theta})$

## General setup: Z-estimators

We observe  $N$  data points  $d_1, \dots, d_N$  (in any domain).



# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\hat{\theta} := \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - \theta^T x_n)^2$$

$$\Leftrightarrow \sum_{n=1}^N (y_n - \hat{\theta}^T x_n) x_n = 0.$$

Make a qualitative decision using:

- A particular component:  $\hat{\theta}_k$
- The end of a confidence interval:  $\hat{\theta}_k + \frac{1.96}{\sqrt{N}} \hat{\sigma}(\hat{\theta})$

## General setup: Z-estimators

We observe  $N$  data points  $d_1, \dots, d_N$  (in any domain).

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\sum_{n=1}^N G(\hat{\theta}, d_n) = 0_P.$$

$G(\cdot, d_n)$  is “nice,”  $\mathbb{R}^P$ -valued.  
E.g. MLE, MAP, VB, IV &c.

# Formalizing the question.

## Example: Least squares

A data point  $d_n$  has regressors  $x_n$  and response  $y_n$ :  $d_n = (x_n, y_n)$ .

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\hat{\theta} := \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - \theta^T x_n)^2$$

$$\Leftrightarrow \sum_{n=1}^N (y_n - \hat{\theta}^T x_n) x_n = 0.$$

Make a qualitative decision using:

- A particular component:  $\hat{\theta}_k$
- The end of a confidence interval:  $\hat{\theta}_k + \frac{1.96}{\sqrt{N}} \hat{\sigma}(\hat{\theta})$

## General setup: Z-estimators

We observe  $N$  data points  $d_1, \dots, d_N$  (in any domain).

The estimator  $\hat{\theta} \in \mathbb{R}^P$  satisfies:

$$\sum_{n=1}^N G(\hat{\theta}, d_n) = 0_P.$$

$G(\cdot, d_n)$  is “nice,”  $\mathbb{R}^P$ -valued.  
E.g. MLE, MAP, VB, IV &c.

Make a qualitative decision using  $\phi(\hat{\theta})$  for a smooth, real-valued  $\phi$ .

(WLOG try to increase  $\phi(\hat{\theta})$ .)

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ?

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )
- Evaluating  $\hat{\theta}$  re-solving the estimating equation.
  - E.g., re-running OLS. ( $0.001s \cdot 1.5 \cdot 10^{51} \approx 4.8 \cdot 10^{40}$  years)
  - Other examples are even harder (VB, machine learning)

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )
- Evaluating  $\hat{\theta}$  re-solving the estimating equation.
  - E.g., re-running OLS. ( $0.001s \cdot 1.5 \cdot 10^{51} \approx 4.8 \cdot 10^{40}$  years)
  - Other examples are even harder (VB, machine learning)

**Our idea:** Smoothly approximate the effect of leaving out points.

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )
- Evaluating  $\hat{\theta}$  re-solving the estimating equation.
  - E.g., re-running OLS. ( $0.001s \cdot 1.5 \cdot 10^{51} \approx 4.8 \cdot 10^{40}$  years)
  - Other examples are even harder (VB, machine learning)

**Our idea:** Smoothly approximate the effect of leaving out points.

---

We have  $N$  data points  $d_1, \dots, d_N$ , a quantity of interest  $\phi(\cdot)$ , and

$$\sum_{n=1}^N G(\hat{\theta}, d_n) = 0_P \quad .$$



# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )
- Evaluating  $\hat{\theta}$  re-solving the estimating equation.
  - E.g., re-running OLS. ( $0.001s \cdot 1.5 \cdot 10^{51} \approx 4.8 \cdot 10^{40}$  years)
  - Other examples are even harder (VB, machine learning)

**Our idea:** Smoothly approximate the effect of leaving out points.

---

We have  $N$  data points  $d_1, \dots, d_N$ , a quantity of interest  $\phi(\cdot)$ , and

$$\sum_{n=1}^N w_n G(\hat{\theta}(w), d_n) = 0_P \text{ for a weight vector } w \in \mathbb{R}^N.$$

# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )
- Evaluating  $\hat{\theta}$  re-solving the estimating equation.
  - E.g., re-running OLS. ( $0.001s \cdot 1.5 \cdot 10^{51} \approx 4.8 \cdot 10^{40}$  years)
  - Other examples are even harder (VB, machine learning)

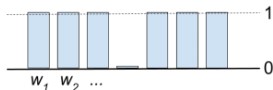
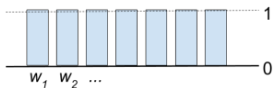
**Our idea:** Smoothly approximate the effect of leaving out points.

---

We have  $N$  data points  $d_1, \dots, d_N$ , a quantity of interest  $\phi(\cdot)$ , and

$$\sum_{n=1}^N w_n G(\hat{\theta}(w), d_n) = 0_P \text{ for a weight vector } w \in \mathbb{R}^N.$$

Original weights:  $\vec{1} = (1, \dots, 1)$  Leave points out by setting their elements of  $w$  to zero.



# Data dropping as data reweighting.

**Question:** Can we make a big change in  $\phi(\hat{\theta})$  by dropping  $\lfloor \alpha N \rfloor$  datapoints, for some small proportion  $\alpha$ ? **Two big problems:**

- There are  $\binom{N}{\lfloor \alpha N \rfloor}$  sets to check. (E.g.  $\binom{16,560}{15} \approx 1.5 \cdot 10^{51}$ )
- Evaluating  $\hat{\theta}$  re-solving the estimating equation.
  - E.g., re-running OLS. ( $0.001s \cdot 1.5 \cdot 10^{51} \approx 4.8 \cdot 10^{40}$  years)
  - Other examples are even harder (VB, machine learning)

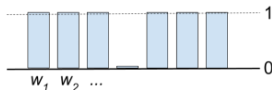
**Our idea:** Smoothly approximate the effect of leaving out points.

---

We have  $N$  data points  $d_1, \dots, d_N$ , a quantity of interest  $\phi(\cdot)$ , and

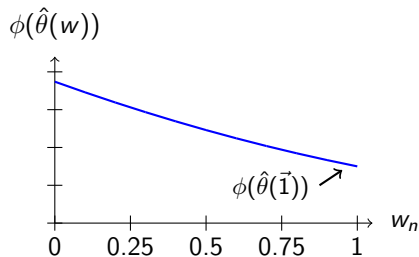
$$\sum_{n=1}^N w_n G(\hat{\theta}(w), d_n) = 0_P \text{ for a weight vector } w \in \mathbb{R}^N.$$

Original weights:  $\vec{1} = (1, \dots, 1)$  Leave points out by setting their elements of  $w$  to zero.

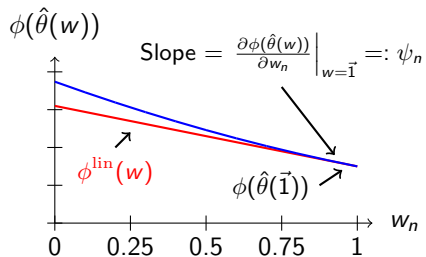


The map  $w \mapsto \phi(\hat{\theta}(w))$  is well-defined even for continuous weights.

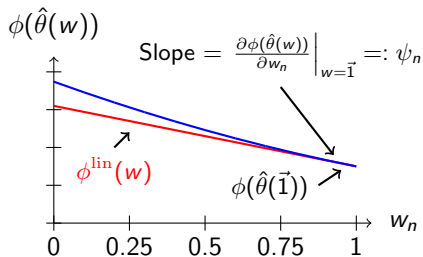
# Taylor series approximation.



# Taylor series approximation.

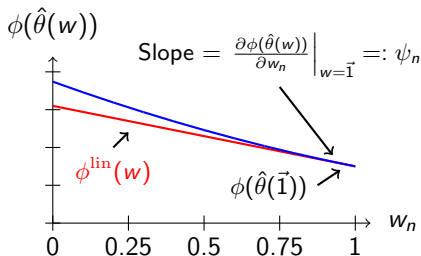


# Taylor series approximation.



The values  $N\psi_n$  are the **empirical influence function** [Hampel, 1986].  
We call  $\psi_n$  an “influence score.”

# Taylor series approximation.



The values  $N\psi_n$  are the **empirical influence function** [Hampel, 1986]. We call  $\psi_n$  an “influence score.”

We can use  $\psi_n$  to form a Taylor series approximation:

$$\phi(\hat{\theta}(w)) \approx \phi^{\text{lin}}(w) := \phi(\hat{\theta}(\vec{1})) + \sum_{n=1}^N \psi_n (w_n - 1)$$

# Taylor series approximation.

**Problem:** How much can you change  $\phi(\hat{\theta}(w))$  dropping  $\lfloor \alpha N \rfloor$  points?  
**Combinatorially hard by brute force!**

---



# Taylor series approximation.

**Problem:** How much can you change  $\phi(\hat{\theta}(w))$  dropping  $\lfloor \alpha N \rfloor$  points?  
**Combinatorially hard by brute force!**

---

**Approximate Problem:** How much can you change  $\phi^{\text{lin}}(w)$  dropping  $\lfloor \alpha N \rfloor$  points? **Easy!**

$$\phi^{\text{lin}}(w) := \phi(\hat{\theta}(\vec{1})) + \sum_{n=1}^N \psi_n(w_n - 1)$$

Dropped points have  $w_n - 1 = -1$ . Kept points have  $w_n - 1 = 0$   
 $\Rightarrow$  The most influential points for  $\phi^{\text{lin}}(w)$  have the most negative  $\psi_n$ .

---

# Taylor series approximation.

**Problem:** How much can you change  $\phi(\hat{\theta}(w))$  dropping  $\lfloor \alpha N \rfloor$  points?  
**Combinatorially hard by brute force!**

---

**Approximate Problem:** How much can you change  $\phi^{\text{lin}}(w)$  dropping  $\lfloor \alpha N \rfloor$  points? **Easy!**

$$\phi^{\text{lin}}(w) := \phi(\hat{\theta}(\vec{1})) + \sum_{n=1}^N \psi_n(w_n - 1)$$

Dropped points have  $w_n - 1 = -1$ . Kept points have  $w_n - 1 = 0$   
 $\Rightarrow$  The most influential points for  $\phi^{\text{lin}}(w)$  have the most negative  $\psi_n$ .

---

**Our procedure:** (see `rgiordan/zaminfluence` on github)

- 1 Compute your original estimator  $\hat{\theta}(\vec{1})$ .
- 2 Compute and sort the influence scores  $\psi_{(1)}, \dots, \psi_{(N)}$ .
- 3 Check if  $-\sum_{n=1}^{\lfloor \alpha N \rfloor} \psi_{(n)}$  is large enough to change your conclusions.

# Taylor series approximation.

**Problem:** How much can you change  $\phi(\hat{\theta}(w))$  dropping  $\lfloor \alpha N \rfloor$  points?  
**Combinatorially hard by brute force!**

---

**Approximate Problem:** How much can you change  $\phi^{\text{lin}}(w)$  dropping  $\lfloor \alpha N \rfloor$  points? **Easy!**

$$\phi^{\text{lin}}(w) := \phi(\hat{\theta}(\vec{1})) + \sum_{n=1}^N \psi_n(w_n - 1)$$

Dropped points have  $w_n - 1 = -1$ . Kept points have  $w_n - 1 = 0$   
 $\Rightarrow$  The most influential points for  $\phi^{\text{lin}}(w)$  have the most negative  $\psi_n$ .

---

**Our procedure:** (see `rgiordan/zaminfluence` on github)

- 1 Compute your original estimator  $\hat{\theta}(\vec{1})$ .
- 2 Compute and sort the influence scores  $\psi_{(1)}, \dots, \psi_{(N)}$ .
- 3 Check if  $-\sum_{n=1}^{\lfloor \alpha N \rfloor} \psi_{(n)}$  is large enough to change your conclusions.

How to compute the  $\psi_n$ 's? And how accurate is the approximation?

# How to compute the influence scores?

How can we compute the influence scores  $\psi_n = \left. \frac{\partial \phi(\hat{\theta}(w))}{\partial w_n} \right|_{w=\vec{1}}$ ?

# How to compute the influence scores?

How can we compute the influence scores  $\psi_n = \left. \frac{\partial \phi(\hat{\theta}(w))}{\partial w_n} \right|_{w=\vec{1}}$ ?

By the **chain rule**,  $\psi_n = \left. \frac{\partial \phi(\theta)}{\partial \theta} \right|_{\hat{\theta}(\vec{1})} \left. \frac{\partial \hat{\theta}(w)}{\partial w_n} \right|_{w=\vec{1}}$ .

# How to compute the influence scores?

How can we compute the influence scores  $\psi_n = \left. \frac{\partial \phi(\hat{\theta}(w))}{\partial w_n} \right|_{w=\vec{1}}$ ?

By the **chain rule**,  $\psi_n = \left. \frac{\partial \phi(\theta)}{\partial \theta} \right|_{\hat{\theta}(\vec{1})} \left. \frac{\partial \hat{\theta}(w)}{\partial w_n} \right|_{w=\vec{1}}$ .

Recall that  $\sum_{n=1}^N w_n G(\hat{\theta}(w), d_n) = 0_P$  for all  $w$  near  $\vec{1}$ .

$\Rightarrow$  By the **implicit function theorem**, we can write  $\left. \frac{\partial \hat{\theta}(w)}{\partial w_n} \right|_{w=\vec{1}}$  as a linear system involving  $G(\cdot, \cdot)$  and its derivatives.

# How to compute the influence scores?

How can we compute the influence scores  $\psi_n = \left. \frac{\partial \phi(\hat{\theta}(w))}{\partial w_n} \right|_{w=\vec{1}}$ ?

By the **chain rule**,  $\psi_n = \left. \frac{\partial \phi(\theta)}{\partial \theta} \right|_{\hat{\theta}(\vec{1})} \left. \frac{\partial \hat{\theta}(w)}{\partial w_n} \right|_{w=\vec{1}}$ .

Recall that  $\sum_{n=1}^N w_n G(\hat{\theta}(w), d_n) = 0_P$  for all  $w$  near  $\vec{1}$ .

$\Rightarrow$  By the **implicit function theorem**, we can write  $\left. \frac{\partial \hat{\theta}(w)}{\partial w_n} \right|_{w=\vec{1}}$  as a linear system involving  $G(\cdot, \cdot)$  and its derivatives.

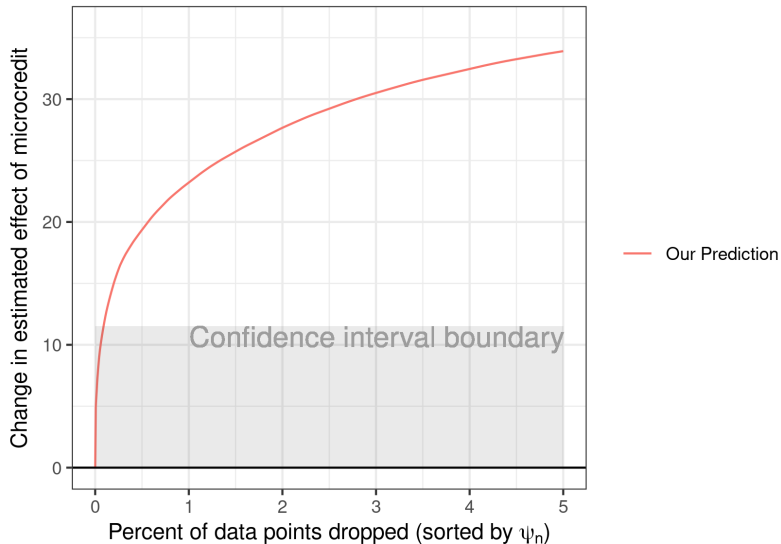
$\Rightarrow$  The  $\psi_n$  are automatically computable from  $\hat{\theta}(\vec{1})$  and software implementations of  $G(\cdot, \cdot)$  and  $\phi(\cdot)$  using **automatic differentiation**.

```
> import jax
> import jax.numpy as np
> def phi(theta):
>     ... computations using np and theta ...
>     return value
>
> # Exact gradient of phi (first term in the chain rule above):
> jax.grad(phi)(theta_opt)
```

See [rgiordan/vittles](#) on github.

# How accurate is the approximation?

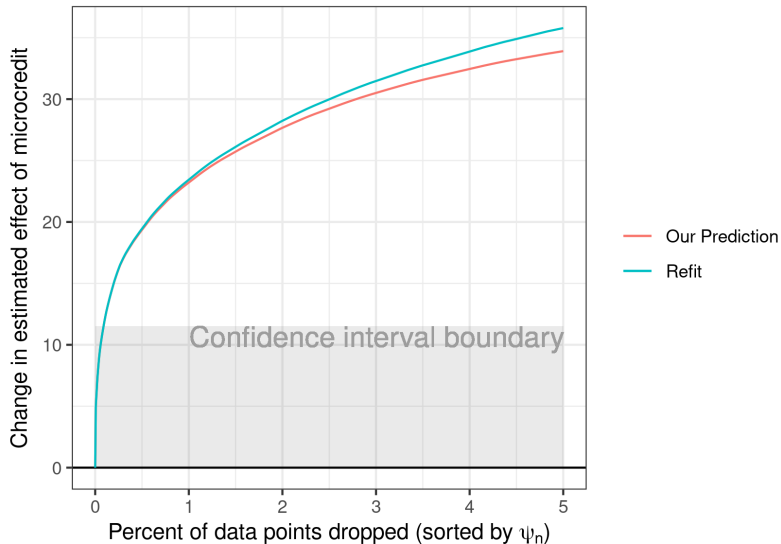
Checking the approximation for Mexico microcredit.





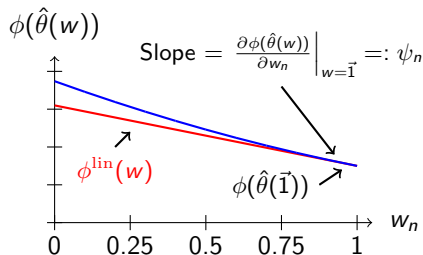
# How accurate is the approximation?

Checking the approximation for Mexico microcredit.



# How accurate is the approximation?

By controlling the curvature, we can control the error in the linear approximation.

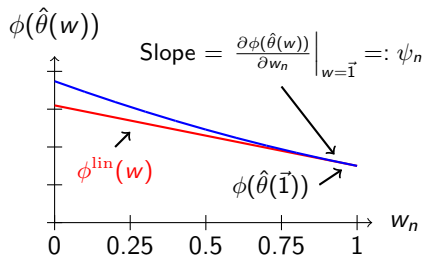


We provide **finite-sample theory** [Giordano et al., 2019b] showing that

$$\left| \phi(\hat{\theta}(w)) - \phi^{\text{lin}}(w) \right| = O \left( \left\| \frac{1}{N}(w - \vec{1}) \right\|_2^2 \right) = O(\alpha) \text{ as } \alpha \rightarrow 0.$$

# How accurate is the approximation?

By controlling the curvature, we can control the error in the linear approximation.



We provide **finite-sample theory** [Giordano et al., 2019b] showing that  $\left| \phi(\hat{\theta}(w)) - \phi^{\text{lin}}(w) \right| = O\left(\left\| \frac{1}{N}(w - \vec{1}) \right\|_2^2\right) = O(\alpha)$  as  $\alpha \rightarrow 0$ .

---

**But you don't need to rely on the theory to show non-robustness!**

Our method returns which points to drop. **Re-running once** without those points provides an **exact lower bound** on the worst-case sensitivity.

## Selected experimental results.

Original estimate (SE)	Refit estimate (SE)	Observations dropped
-4.549 (5.879)	7.030 (2.549)*	15 = 0.09%

Table: Microcredit Mexico results ( $N = 16560$ ) [Angelucci et al., 2015].

A \* indicates statistical significance at the 95% level.

## Selected experimental results.

Original estimate (SE)	Refit estimate (SE)	Observations dropped
-4.549 (5.879)	7.030 (2.549)*	15 = 0.09%

Table: Microcredit Mexico results (N = 16560) [Angelucci et al., 2015].

Original estimate (SE)	Refit estimate (SE)	Observations dropped
33.861 (4.468)*	-9.416 (3.296)*	986 = 9.37%

Table: Cash transfers results (N = 10518) [Angelucci and De Giorgi, 2009]

A \* indicates statistical significance at the 95% level.

## Selected experimental results.

Original estimate (SE)	Refit estimate (SE)	Observations dropped
-4.549 (5.879)	7.030 (2.549)*	15 = 0.09%

Table: Microcredit Mexico results (N = 16560) [Angelucci et al., 2015].

Original estimate (SE)	Refit estimate (SE)	Observations dropped
33.861 (4.468)*	-9.416 (3.296)*	986 = 9.37%

Table: Cash transfers results (N = 10518) [Angelucci and De Giorgi, 2009]

Original estimate (SE)	Refit estimate (SE)	Observations dropped
0.029 (0.005)*	-0.009 (0.004)*	224 = 0.96%

Table: Medicaid profit results (N = 23361) [Finkelstein et al., 2012]

A \* indicates statistical significance at the 95% level.

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1}))$$

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1}))$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion



# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) =: \hat{\sigma}_\phi \hat{\mathcal{S}}_\alpha$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) =: \hat{\sigma}_\phi \hat{\mathcal{J}}_\alpha$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion
- The “noise”  $\hat{\sigma}_\phi^2 \rightarrow \lim_{N \rightarrow \infty} \text{Var}(\sqrt{N}\phi)$  (“sandwich” variance estimator)

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) =: \hat{\sigma}_\phi \hat{\mathcal{J}}_\alpha$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion
- The “noise”  $\hat{\sigma}_\phi^2 \rightarrow \lim_{N \rightarrow \infty} \text{Var}(\sqrt{N}\phi)$  (“sandwich” variance estimator)
- The “shape”  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant and is  $\leq \sqrt{\alpha(1-\alpha)}$

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) =: \hat{\sigma}_\phi \hat{\mathcal{J}}_\alpha \quad \Leftrightarrow \quad \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha.$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion
- The “noise”  $\hat{\sigma}_\phi^2 \rightarrow \lim_{N \rightarrow \infty} \text{Var}(\sqrt{N}\phi)$  (“sandwich” variance estimator)
- The “shape”  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant and is  $\leq \sqrt{\alpha(1-\alpha)}$

---

The **signal to noise ratio**  $\frac{\Delta}{\hat{\sigma}_\phi}$  determines robustness to data dropping ...

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) =: \hat{\sigma}_\phi \hat{\mathcal{J}}_\alpha \quad \Leftrightarrow \quad \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha.$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion
- The “noise”  $\hat{\sigma}_\phi^2 \rightarrow \lim_{N \rightarrow \infty} \text{Var}(\sqrt{N}\phi)$  (“sandwich” variance estimator)
- The “shape”  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant and is  $\leq \sqrt{\alpha(1-\alpha)}$

---

The **signal to noise ratio**  $\frac{\Delta}{\hat{\sigma}_\phi}$  determines robustness to data dropping ...

---

## Contrast with sampling variability.

A 95% CI is given by  $\phi(\hat{\theta}(\vec{1})) \pm \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ . We reject  $\phi(\hat{\theta}(\vec{1})) + \Delta$  when

$$\phi(\hat{\theta}(\vec{1})) + \Delta \geq \phi(\hat{\theta}(\vec{1})) + \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$$

# What makes an analysis sensitive? Preliminaries

We are **robust to data dropping** if, for the  $\Delta$  that changes conclusions and  $w^*$  dropping the  $\lfloor \alpha N \rfloor$  most influential points,

$$\Delta \geq \phi^{\text{lin}}(w^*) - \phi(\hat{\theta}(\vec{1})) =: \hat{\sigma}_\phi \hat{\mathcal{J}}_\alpha \quad \Leftrightarrow \quad \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha.$$

- The “signal”  $\Delta$  is the smallest change that reverses your conclusion
- The “noise”  $\hat{\sigma}_\phi^2 \rightarrow \lim_{N \rightarrow \infty} \text{Var}(\sqrt{N}\phi)$  (“sandwich” variance estimator)
- The “shape”  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant and is  $\leq \sqrt{\alpha(1-\alpha)}$

---

The **signal to noise ratio**  $\frac{\Delta}{\hat{\sigma}_\phi}$  determines robustness to data dropping **and** sampling variability, but with **different thresholds**.

---

## Contrast with sampling variability.

A 95% CI is given by  $\phi(\hat{\theta}(\vec{1})) \pm \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ . We reject  $\phi(\hat{\theta}(\vec{1})) + \Delta$  when

$$\phi(\hat{\theta}(\vec{1})) + \Delta \geq \phi(\hat{\theta}(\vec{1})) + \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi \quad \Leftrightarrow \quad \frac{\Delta}{\hat{\sigma}_\phi} \geq \frac{1.96}{\sqrt{N}}.$$

# What makes an analysis sensitive?

Robust to data dropping:  
(“dropping robustness”)

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to sampling variation:  
(“sampling robustness”)

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \frac{1.96}{\sqrt{N}}$$

---

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to sampling variation:  
("sampling robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \frac{1.96}{\sqrt{N}}$$

- 
- **Dropping robustness  $\neq$  sampling robustness in general.**

*Proof:*  $\hat{\mathcal{J}}_\alpha \neq \frac{1.96}{\sqrt{N}}$ .



# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to sampling variation:  
("sampling robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \frac{1.96}{\sqrt{N}}$$

- 
- **Dropping robustness  $\neq$  sampling robustness in general.**

*Proof:*  $\hat{\mathcal{J}}_\alpha \neq \frac{1.96}{\sqrt{N}}$ .

- **When the SNR is small, sufficiently large  $N$  produces sampling robustness, but not necessarily dropping robustness.**

*Proof:*  $\frac{1.96}{\sqrt{N}} \rightarrow 0$ , but  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant.

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to sampling variation:  
("sampling robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \frac{1.96}{\sqrt{N}}$$

- 
- **Dropping robustness  $\neq$  sampling robustness in general.**

*Proof:*  $\hat{\mathcal{J}}_\alpha \neq \frac{1.96}{\sqrt{N}}$ .

- **When the SNR is small, sufficiently large  $N$  produces sampling robustness, but not necessarily dropping robustness.**

*Proof:*  $\frac{1.96}{\sqrt{N}} \rightarrow 0$ , but  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant.

- **Statistical insignificance is dropping non-robust for large  $N$ .**

*Proof:* Insignificance means  $|\phi(\hat{\theta}(\vec{1}))| \leq \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ .

$\Rightarrow$  A result can be made significant by a change of no more than  $\frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ .

$\Rightarrow$  The SNR for a conclusion of "insignificance" is  $\frac{\Delta}{\hat{\sigma}_\phi} \leq \frac{1.96}{\sqrt{N}} \rightarrow 0 \leq \hat{\mathcal{J}}_\alpha$ .

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to sampling variation:  
("sampling robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \frac{1.96}{\sqrt{N}}$$

- 
- **Dropping robustness  $\neq$  sampling robustness in general.**

*Proof:*  $\hat{\mathcal{J}}_\alpha \neq \frac{1.96}{\sqrt{N}}$ .

- **When the SNR is small, sufficiently large  $N$  produces sampling robustness, but not necessarily dropping robustness.**

*Proof:*  $\frac{1.96}{\sqrt{N}} \rightarrow 0$ , but  $\hat{\mathcal{J}}_\alpha \rightarrow$  a nonzero constant.

- **Statistical insignificance is dropping non-robust for large  $N$ .**

*Proof:* Insignificance means  $|\phi(\hat{\theta}(\vec{1}))| \leq \frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ .

$\Rightarrow$  A result can be made significant by a change of no more than  $\frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ .

$\Rightarrow$  The SNR for a conclusion of "insignificance" is  $\frac{\Delta}{\hat{\sigma}_\phi} \leq \frac{1.96}{\sqrt{N}} \rightarrow 0 \leq \hat{\mathcal{J}}_\alpha$ .

- **P-hacking is dropping non-robust for large  $N$ .**

*Proof:* P-hacked effect sizes are of the order  $\frac{1.96}{\sqrt{N}} \hat{\sigma}_\phi$ .

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_{\phi}} \geq \hat{\mathcal{J}}_{\alpha}$$

Robust to gross errors:  
("gross error robustness")

Gross outliers cannot produce  
arbitrarily large changes to  $\phi$ .

---

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to gross errors:  
("gross error robustness")

Gross outliers cannot produce  
arbitrarily large changes to  $\phi$ .

- 
- **Dropping non-robustness is not driven by misspecification.**

*Proof:* Small  $\Delta$  are dropping non-robust irrespective of specification.

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to gross errors:  
("gross error robustness")

Gross outliers cannot produce  
arbitrarily large changes to  $\phi$ .

---

- **Dropping non-robustness is not driven by misspecification.**

*Proof:* Small  $\Delta$  are dropping non-robust irrespective of specification.

- **Gross outliers primarily affect dropping robustness through  $\hat{\sigma}_\phi$ .**

*Proof:* For a fixed  $\hat{\sigma}_\phi$ , outliers decrease  $\hat{\mathcal{J}}_\alpha$ . (Details in paper.)

# What makes an analysis sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \hat{\mathcal{J}}_\alpha$$

Robust to gross errors:  
("gross error robustness")

Gross outliers cannot produce  
arbitrarily large changes to  $\phi$ .

- 
- **Dropping non-robustness is not driven by misspecification.**

*Proof:* Small  $\Delta$  are dropping non-robust irrespective of specification.

- **Gross outliers primarily affect dropping robustness through  $\hat{\sigma}_\phi$ .**

*Proof:* For a fixed  $\hat{\sigma}_\phi$ , outliers decrease  $\hat{\mathcal{J}}_\alpha$ . (Details in paper.)

---

Dropping robustness should **augment** other forms of robustness.

# How to make an analysis less sensitive?

Robust to data dropping:  
("dropping robustness")

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \mathcal{J}_\alpha$$

---

**To achieve dropping robustness, reduce  $\hat{\sigma}_\phi$  and / or increase  $\Delta$ .**

*Proof:* Across typical distributions,  $\mathcal{J}_\alpha$  varies little. (Details in paper.)



# How to make an analysis less sensitive?

Robust to data dropping:  
 (“dropping robustness”)

$$\text{SNR} := \frac{\Delta}{\hat{\sigma}_\phi} \geq \mathcal{J}_\alpha$$

---

**To achieve dropping robustness, reduce  $\hat{\sigma}_\phi$  and / or increase  $\Delta$ .**

*Proof:* Across typical distributions,  $\mathcal{J}_\alpha$  varies little. (Details in paper.)

In the Mexico microcredit example,

$$\hat{\sigma}_\phi = 757.8 \qquad \phi(\hat{\theta}(\vec{1})) = -4.55 \qquad N = 16,560$$

The study overcame a very low signal to noise ratio with a very large  $N$ .

This (canonical) response to low signal to noise ratio — to gather more data — produces small SEs, but cannot produce dropping robustness.

- You may be concerned if you could reverse your conclusion by removing a small proportion of your data.

- You may be concerned if you could reverse your conclusion by removing a small proportion of your data.
- We can quickly and automatically find an approximate influential set which is accurate for small sets.

- You may be concerned if you could reverse your conclusion by removing a small proportion of your data.
- We can quickly and automatically find an approximate influential set which is accurate for small sets.
- Data dropping robustness is principally determined by the signal to noise ratio, and captures sensitivity distinct from sampling and gross error sensitivity.

# Links and references

Tamara Broderick, Ryan Giordano, Rachael Meager (alphabetical authors)  
“An Automatic Finite-Sample Robustness Metric: Can Dropping a Little Data Change Conclusions?”

<https://arxiv.org/abs/2011.14999>

---

Select blog posts with more details: <https://rgiordan.github.io>

- Data dropping sensitivity overcomes p-hacking
  - Collinearity in OLS after dropping
  - Influence functions and sums
  - Connections to the bootstrap
- 

Related software on github:

- [rgiordan/zaminfluence](#) (for R)
  - [rgiordan/vittles](#) (for Python)
- 

Some of my work on other forms of robustness:

- Prior sensitivity in Bayesian nonparametrics [Giordano et al., 2021]
- Approximate cross-validation (and other reweightings) [Giordano et al., 2019b,a]
- Covariances and prior sensitivity for mean field VB [Giordano et al., 2015, 2018]
- Model sensitivity of MCMC output [Giordano et al., 2018]
- Frequentist variances of MCMC posteriors (in progress)

- M. Angelucci and G. De Giorgi. Indirect effects of an aid program: How do cash transfers affect ineligible's consumption? *American Economic Review*, 99(1):486–508, 2009.
- M. Angelucci, D. Karlan, and J. Zinman. Microcredit impacts: Evidence from a randomized microcredit program placement experiment by Compartamos Banco. *American Economic Journal: Applied Economics*, 7(1):151–82, 2015.
- P. Bickel, C. Klaassen, Y. Ritov, and J. Wellner. *Efficient and adaptive estimation for semiparametric models*, volume 4. Johns Hopkins University Press Baltimore, 1993.
- A. Finkelstein, S. Taubman, B. Wright, M. Bernstein, J. Gruber, J. Newhouse, H. Allen, K. Baicker, and Oregon Health Study Group. The Oregon health insurance experiment: Evidence from the first year. *The Quarterly Journal of Economics*, 127(3):1057–1106, 2012.
- R. Giordano, T. Broderick, and M. I. Jordan. Linear response methods for accurate covariance estimates from mean field variational Bayes. *Advances in Neural Information Processing Systems*, 28:1441–1449, 2015.
- R. Giordano, T. Broderick, and M. I. Jordan. Covariances, robustness and variational Bayes. *The Journal of Machine Learning Research*, 19(1):1981–2029, 2018.
- R. Giordano, M. I. Jordan, and T. Broderick. A higher-order Swiss army infinitesimal jackknife. *arXiv preprint arXiv:1907.12116*, 2019a.
- R. Giordano, W. Stephenson, R. Liu, M. I. Jordan, and T. Broderick. A Swiss army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147. PMLR, 2019b.
- R. Giordano, R. Liu, M. I. Jordan, and T. Broderick. Evaluating sensitivity to the stick-breaking prior in Bayesian nonparametrics. 2021.
- F. Hampel. *Robust statistics: The approach based on influence functions*, volume 196. Wiley-Interscience, 1986.
- P. Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.
- R. Mises. On the asymptotic distribution of differentiable statistical functions. *The Annals of Mathematical Statistics*, 18(3):309–348, 1947.
- J. Reeds. *On the definition of von Mises functionals*. PhD thesis, Statistics, Harvard University, 1976.

# Extra slides

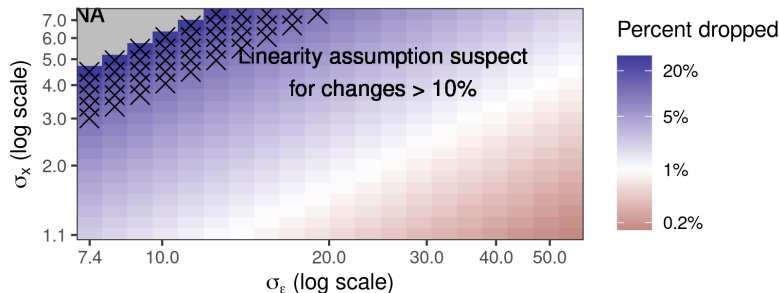
# A simulation

For  $N = 5,000$  data points, compute the OLS estimator from:

Regressors  
 $x_n \sim \mathcal{N}(0, \sigma_x^2)$

Residuals  
 $\varepsilon_n \sim \mathcal{N}(0, \sigma_\varepsilon^2)$

Responses  
 $y_n = 0.5x_n + \varepsilon_n$



**Figure:** The approximate perturbation inducing proportion at differing values of  $\sigma_x$  and  $\sigma_\varepsilon$ . Red colors indicate datasets whose sign can be predicted to change when dropping less than 1% of datapoints. The grey areas indicate  $\hat{\Psi}_\alpha = \text{NA}$ , a failure of the linear approximation to locate any way to change the sign.



# Influence function

The present work is based on the *empirical influence function*. Consider:

- True, unknown distribution function  $F_\infty(x) = p(X \leq x)$
- Empirical distribution function  $\hat{F}(x) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(x_n \leq x)$
- A statistical functional  $T(F)$ .

# Influence function

The present work is based on the *empirical influence function*. Consider:

- True, unknown distribution function  $F_\infty(x) = p(X \leq x)$
- Empirical distribution function  $\hat{F}(x) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(x_n \leq x)$
- A statistical functional  $T(F)$ .

We estimate with  $T(F_\infty)$  with  $T(\hat{F})$ .

Sample means are an example:

$$T(F) := \int x F(dx).$$

Z-estimators are, too:

$$T(F) := \theta \text{ such that } \int G(\theta, x) F(dx) = 0.$$

# Influence function

The present work is based on the *empirical influence function*. Consider:

- True, unknown distribution function  $F_\infty(x) = p(X \leq x)$
- Empirical distribution function  $\hat{F}(x) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(x_n \leq x)$
- A statistical functional  $T(F)$ .

Form an (infinite-dimensional) Taylor series expansion at some  $F_0$ :

$$T(F) = T(F_0) + T'(F_0)(F - F_0) + \text{residual}.$$

When the derivative operator takes the form of an integral

$$T'(F_0)\Delta = \int \psi(x; F_0)\Delta(dx)$$

then  $\psi(x; F_0)$  is known as the *influence function*.

Where to form the expansion? There are at least two reasonable choices:

- The limiting influence function  $\psi(x, F_\infty)$
- The empirical influence function  $\psi(x, \hat{F})$

# Influence function

- The limiting influence function (LIF)  $\psi(x, F_\infty)$ 
  - Used in a lot of classical statistics [Mises, 1947, Huber, 1981, Hampel, 1986, Bickel et al., 1993]
  - Unobserved, asymptotic
  - Requires careful functional analysis [Reeds, 1976]
- The empirical influence function (EIF)  $\psi(x, \hat{F})$ 
  - The basis of the present work (also [Giordano et al., 2019b,a])
  - Computable, finite-sample
  - Requires only finite-dimensional calculus

Typically the *semantics* of the EIF derive from study of the LIF.

Example:  $\frac{1}{N} \sum_{n=1}^N (N\psi_n)^2 \approx \text{Var} \left( \sqrt{N}\phi(\hat{\theta}) \right).$

But the EIF measures what happens when you perturb the data at hand.

Other data perturbations will admit an analysis similar to ours!

The present work is an application of *local robustness*. Consider:

- Model parameter  $\lambda$  (e.g., data weights  $\lambda = w$ )
- Set of plausible models  $\mathcal{S}_\lambda$  (e.g.  $\mathcal{S}_\lambda = W_\alpha$ )
- Estimator  $\hat{\theta}(x, \lambda)$  for data  $x$  and  $\lambda \in \mathcal{S}_\lambda$  (e.g. a Z-estimator)

---

Global robustness:  $\left( \inf_{\lambda \in \mathcal{S}_\lambda} \hat{\theta}(x, \lambda), \sup_{\lambda \in \mathcal{S}_\lambda} \hat{\theta}(x, \lambda) \right)$  (Hard in general!)

---

Local robustness:  $\left( \inf_{\lambda \in \mathcal{S}_\lambda} \hat{\theta}^{lin}(x, \lambda), \sup_{\lambda \in \mathcal{S}_\lambda} \hat{\theta}^{lin}(x, \lambda) \right)$

---

...where  $\hat{\theta}^{lin}(x, \lambda) := \hat{\theta}^{lin}(x, \lambda_0) + \left. \frac{\partial \hat{\theta}^{lin}(x, \lambda)}{\partial \lambda} \right|_{\lambda_0} (\lambda - \lambda_0)$ .

---

**Many variants are possible!**

- Cross-validation [Giordano et al., 2019b]
- Prior sensitivity in Bayesian nonparametrics [Giordano et al., 2021]
- Model sensitivity of MCMC output [Giordano et al., 2018]
- Frequentist variances of MCMC posteriors (in progress)